IN THE CLAIMS:

1. (Currently Amended) A computer-implemented method for programmatically generating a graphical program based on a state diagram, comprising:

receiving state diagram information, wherein the state diagram information represents the state diagram and specifies [[one or more]] a plurality of states;

programmatically generating the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the [[specified one or more]] plurality of states, wherein the graphical source code comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and wherein the graphical program is executable by a computer.

2. (Cancelled)

3. (Currently Amended) The method of claim [[2]] 1,
wherein the state diagram represents desired operation of a software program.

4. (Currently Amended) The method of claim [[2]] 1,
wherein the state diagram represents desired operation of a hardware device.

5. (Currently Amended) The method of claim [[2]] 1,
wherein the state diagram represents a desired algorithm.

6. (Currently Amended) The method of claim [[2]] 1,
wherein the state diagram represents a test sequence.

7. (Currently Amended) The method of claim 1,
wherein said programmatically generating the [[new]] graphical program creates the [[new]] graphical program without any user input specifying the [[new]] graphical program during said creating.

2

8. (Original) The method of claim 1,

wherein said programmatically generating the graphical program comprises programmatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

9-11. (Cancelled)

12. (Original) The method of claim 1,

wherein, for at least one state, the state diagram information specifies program code associated with the state;

wherein the programmatically generated graphical source code includes the specified program code.

13. (Original) The method of claim 1,

wherein, for at least one state, the state diagram information specifies program code associated with the state;

wherein the programmatically generated graphical source code is operable to invoke the specified source code.

14. (Original) The method of claim 1,

wherein the state diagram information further specifies one or more state transitions, wherein each state transition specifies a transition from a first state to a second state;

wherein said programmatically generating further comprises programmatically generating graphical source code corresponding to the specified state transitions.

15. (Original) The method of claim 14,

wherein the programmatically generated graphical source code includes placeholder graphical source code for each state transition.

16. (Original) The method of claim 15, further comprising:

for one or more state transitions, a user manually entering graphical source code specifying a Boolean condition associated with the state transition.

17. (Original) The method of claim 14,

wherein the state diagram information specifies at least two state transitions from a particular state;

wherein the state diagram information also specifies a priority ordering for the at least two state transitions;

wherein said programmatically generating comprises programmatically generating graphical source code such that, during execution of the graphical program, Boolean conditions associated with the at least two state transitions are evaluated in the specified priority ordering.

18. (Original) The method of claim 1,

wherein the state diagram information further specifies an initially active state;

wherein said programmatically generating comprises programmatically generating graphical source code such that the graphical program begins execution in the initially active state.

19. (Original) The method of claim 1,

wherein the state diagram information further specifies one or more stop states;

wherein said programmatically generating comprises programmatically generating graphical source code such that if during execution of the graphical program one of the stop states becomes active, then the graphical program is caused to stop execution.

20. (Original) The method of claim 1, further comprising:

receiving information specifying a change to the state diagram information;

programmatically updating the graphical program to reflect the specified change.

21. (Original) The method of claim 1,

wherein said programmatically generating the graphical program comprises calling an application programming interface (API) enabling the programmatic generation of a graphical program.

22. (Original) The method of claim 1,

wherein said programmatically generating the graphical program comprises programmatically requesting a server program to generate the graphical program.

23. (Currently Amended) A computer-implemented method for programmatically generating a new graphical program, comprising:

receiving information specifying a state diagram, wherein the state diagram specifies first functionality;

executing a graphical program generation (GPG) program;

the GPG program programmatically generating the new graphical program using said information, wherein the new graphical program includes graphical source code corresponding to the state diagram, wherein the new graphical program comprises a plurality of interconnected nodes which visually indicate operation of the graphical program, and wherein the new graphical program is executable by a computer to perform the first functionality.

24. (Original) The method of claim 23, wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

25. (Currently Amended) A computer-implemented method for programmatically generating a graphical program based on a state diagram, comprising:

displaying an initial state diagram;

5

programmatically generating a graphical program corresponding to the initial state diagram, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and wherein the graphical program is executable by a computer;

receiving user input specifying a change to the initial state diagram;

programmatically updating the graphical program to correspond to the specified change, in response to the user input specifying the change.

26. (Currently Amended) A system for programmatically generating a graphical program, the system comprising:

a processor coupled to a memory, wherein the memory stores a graphical program generation (GPG) program;

wherein the processor is operable to execute the GPG program in order to:

receive state diagram information, wherein the state diagram information specifies one or more states;

programmatically generate the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the specified one or more states, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality.

27. (Original) The system of claim 26,

wherein said programmatically generating the graphical program creates the graphical program without any user input specifying the graphical program during said creating.

28. (Original) The system of claim 26,

6

wherein said programmatically generating the graphical program comprises programmatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

29. (Currently Amended) A memory medium for programmatically generating a graphical program based on a state diagram, the memory medium comprising program instructions executable to:

receive state diagram information, wherein the state diagram information specifies [[one or more]] a plurality of states and one or more transitions between the plurality of states;

programmatically generate the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the specified one or more states, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate operation of the graphical program, and wherein the graphical program is executable by a computer to perform functionality specified by the state diagram information.

30. (Original) The memory medium of claim 29,

wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

31. (Original) The memory medium of claim 29,

wherein said programmatically generating the graphical program comprises programmatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

32. (New) A computer-implemented method for programmatically generating a graphical program based on state diagram information, comprising:

7

receiving the state diagram information, wherein the state diagram information specifies a plurality of states and transitions between the states;

programmatically generating the graphical program in response to the state diagram information, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein a first one or more nodes comprise graphical source code executable to implement first functionality corresponding to a first one or more states, and wherein a second one or more nodes are user-configurable to implement second functionality of a corresponding second one or more states.

33. (New) The method of claim 32,

wherein the programmatically generated graphical program includes placeholder graphical source code for each of the second one or more states.

34. (New) The method of claim 33, further comprising:

for each of the second one or more nodes, a user manually entering graphical source code specifying execution instructions to be performed when the state is active during execution of the graphical program.

35. (New) The method of claim 33,

wherein the placeholder graphical source code for each state comprises a case in a graphical case structure.